Contents

0-8493-0052-5/00/$0.00+$.50

List of Tables

0-8493-0052-5/00/$0.00+$.50

List of Figures

0-8493-0052-5/00/$0.00+$.50

Part I

This is a Part

Chapter 1

A Bayesian feature selection score based on Naive Bayes models

Susana Eyheramendy

Ludwig-Maximilians Universität München

David Madigan

Rutgers University

1.1 Introduction

The past decade has seen the emergence of truly massive data analysis challenges across a range of human endeavors. Standard statistical algorithms came into being long before such challenges were even imagined and, spurred on by myriad important applications, much statistical research now focuses on the development of algorithms that scale well. Feature selection represent a central issue on this research.

Feature selection addresses scalability by removing irrelevant, redundant or noisy features. Feature or variable selection has been applied to many different problems for many different purposes. For example, in text categorization problems, feature selection is often applied to select a subset of relevant words that appear in documents. This can help to elucidate the category or class of unobserved documents. Another area of application that is becoming popular is in the area of genetic association studies, where the aim is to try to find genes responsible for a particular disease (e.g. [14]). In those studies, hundreds of thousands or even a couple of million positions in the genome are genotyped in individuals who have the disease and individuals who do not have the disease. Feature selection in this context seeks to reduce the genotyping of correlated positions in order to decrease the genotyping cost while still being able to find the genes responsible for a given disease.

Feature selection is an important step in the preprocessing of the data. Removing irrelevant and noisy features helps generalization performance, and in addition reduces the computational cost and the memory demands. Reducing the number of variables can also aid in the interpretation of data and in the better distribution of resources.

0-8493-0052-5/00/$0.00+$.50

In this study, we introduce a new feature selection method for classification problems. In particular, we apply our novel method to text categorization problems and compare its performance with other prominent feature selection methods popular in the field of text categorization.

Since many text classification applications involve large numbers of candidate features, feature selection algorithms play a fundamental role. The text classification literature tends to focus on feature selection algorithms that compute a score independently for each candidate feature. This is the so-called *filtering* approach. The scores typically contrast the counts of occurrences of words or other linguistic artifacts in training documents that belong to the target class with the same counts for documents that do not belong to the target class. Given a predefined number of words to be selected, d, one chooses the d words with the highest scores. Several score functions exist (Section 1.2 provides definitions). [16] show that Information Gain and χ^2 statistics performed best among five different scores. [5] provides evidence that these two scores have correlated failures. Hence, when choosing optimal pairs of scores these two scores work poorly together. [5] introduced a new score, the Bi-Normal Separation, that yields the best performance on the greatest number of tasks among twelve feature selection scores. [13] compare eleven scores under a Naive Bayes classifier and find that the Odds Ratio score performes best in the highest number of tasks.

In regression and classification problems in Statistics, popular feature selection strategies depend on the same algorithm that fits the models. This is the so-called *wrapper* approach. For example, *Best subset regression* finds for each k the best subset of size k based on residual sum of squares. *Leaps and bounds* is an efficient algorithm that finds the best set of features when the number of predictors is no larger than about 40. An extensive discussion on subset selection on regression problems is provided by [12]. The recent paper [10] gives a detailed categorization of all existing feature selection methods.

In a Bayesian context and under certain assumptions, [1] shows that for selection among Normal linear models, the best model contains those features which have overall posterior probability greater than or equal to 1/2. Motivated by this study, we introduce a new feature selection score (PIP) that evaluates the posterior probability of inclusion of a given feature over all possible models, where the models correspond to a set of features. Unlike typical scores used for feature selection via filtering, the PIP score *does* depend on a specific model. In this sense, this new score straddles the filtering and wrapper approaches.

We present experiments that compare our new feature selection score with five other feature selection scores that have been prominent in the studies mentioned above. The feature selection scores that we consider are evaluated on two widely-used benchmark text classification datasets, Reuters-21578 and 20-Newsgroups, and implemented on four classification algorithms. Following previous studies, we measure the performance of the classification algorithms using the F_1 measure.

TABLE 1.1:
Two-way contingency table of word F and category k

	k	$\overline{k}$	
F	n_{kF}	$n_{\overline{k}F}$	n_F
$\overline{F}$	$n_{k\overline{F}}$	$n_{\overline{kF}}$	$n_{\overline{F}}$
	n_k	$n_{\overline{k}}$	M

We have organized this chapter as follows. Section 1.2 describes the various feature selection scores we consider, both our new score and the various existing alternatives. In Section 1.3 we mention the classification algorithms that we use to compare the feature selection scores. The experimental settings and experimental results are presented in Section 1.4. We conclude in Section 1.5.

1.2 Feature Selection Scores

In this section we introduce a new methodology to define a feature score and review the definition of other popular feature selection scores.

Before we list the feature selection scores that we study, we introduce some notation. In the context of our text categorization application, Table 1.1 show the basic statistics for a single word and a single category (or class).

n_{kF} : n° of documents in class k with word F.
$n_{k\overline{F}}$: n° of documents in class k without word F.
$n_{\overline{k}F}$: n° of documents not in class k with word F.
$n_{\overline{kF}}$: n° of documents not in class k without word F.
n_k : total n° of documents in class k.
$n_{\overline{k}}$: total n° of documents that are not in class k.
n_F : total n° of documents with word F.
$n_{\overline{F}}$: total n° of documents without word F.
M : total n° of documents.

We refer to F as a word or feature occuring in documents and x as the value that depends on the number of times the word F appears in a document. For example, consider a document that consists on the phrase "curiosity begets curiosity". If F_1 represents the word "curiosity", then x_1 can take the value 1 if we consider the presence or absence of the words in the documents, or x_1 can take the value 2 if the actual frequency of appearance is considered.

1.2.1 Posterior Inclusion Probability (PIP)

Consider a classification problem in which one has M instances in training data $Data = \{(y_1, \mathbf{x}_1), \ldots, (y_M, \mathbf{x}_M)\}$, where y_i denotes the class label of instance i that takes values in a finite set of C classes, and $\mathbf{x}_i$ is its corresponding vector of N features. We consider a Naive Bayes model where the probability of the training data instances can be expressed as the product of the individual conditional probablities of each feature given the class membership, times the probablities of the class memberships,

$$Pr((y_1, \mathbf{x}_1), \ldots, (y_M, \mathbf{x}_M)) = \Pi_{i=1}^{M} Pr(y_i) \Pi_{j=1}^{N} Pr(x_{ij}|y_i). \tag{1.1}$$

We aim to select a subset of the features with which one can infer accurately the class label of new instances using a prediction function or rule that links the vector of features with the class label.

Given N features, one can consider 2^N different models, each one containing a different subset of features. We denote each model by a vector of length the number of features N, where each component is either 1 if the feature is present or 0 if the feature is absent. For two features, Figure 1.1 shows a graphical representation of the four possible models. For example, model $M_{(1,1)}$ contains both features, and the distribution of each feature depends on the class label of the document. This is represented in the graph with an arrow from the node y to each of the features x_1 and x_2.

Without assuming any distribution on the conditional probabilities in equation 1.1, we propose as a feature score the Posterior Inclusion Probability (PIP) for feature F_j and class k, which is defined as

$$PIP(F_j, k) = \sum_{\mathbf{l}: l_j = 1} Pr(M_{\mathbf{l}}|Data), \tag{1.2}$$

where $\mathbf{l}$ is a vector of length the number of features and the jth component takes the value 1 if the jth feature F_j is included in model $M_{\mathbf{l}}$, otherwise it is 0. In other words, we define as the feature selection score, the posterior probability that each feature is included in a model, for all features appearing in documents or instances of class k.

Each feature appears in 2^{N-1} models. For moderate values of N, the sum 1.2 can be extremely large. Fortunately, we show in the next section that it is not necessary to compute the sum 1.2 because it can be expressed in closed form.

Note that for each class, each feature is assigned a different score. The practitioner can either select a different set of features for each of the classes or a single score can be obtained by weighting the scores over all the classes by the frequency of instances in each class. We follow the latter approach in all features selection scores considered in this study. The next two sections implement the PIP feature selection score. The next section assumes a Bernoulli distribution for the conditional probabilities in equation 1.1 and the subsequent section assumes a Poisson distribution.

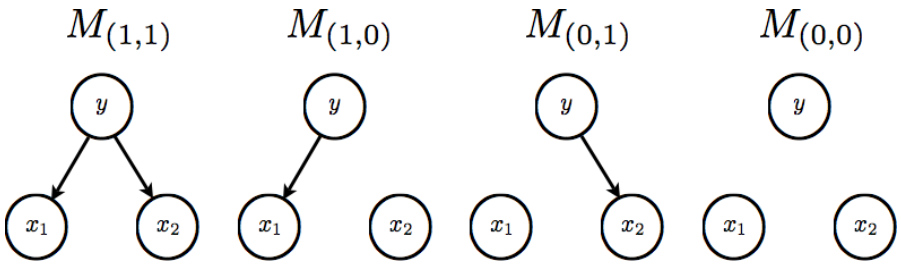

FIGURE 1.1: Graphical model representation of the four models with two features, x_1 and x_2.

1.2.2 Posterior Inclusion Probability (PIP) under a Bernoulli distribution

Consider first that the conditional probabilities $P(x_{ij}|y_i, \theta)$ are Bernoulli distributed with parameter θ, and assume a Beta prior distribution on θ. This is the binary naive Bayes model for the presence or absence of words in the documents. Section 1.2.3 considers a naive Bayes model with Poisson distributions for word frequency. This score for feature F and class k can be expressed as

$$PIP(F,k) = \frac{l_{0Fk}}{l_{0Fk} + l_{Fk}}, \tag{1.3}$$

where

$$l_{0Fk} = \frac{B(n_{kF} + \alpha_{kF}, n_{k\overline{F}}\beta_{kF})}{B(\alpha_{kF}, \beta_{kF})} \tag{1.4}$$

$$\times \frac{B(n_{\overline{k}F} + \alpha_{\overline{k}F}, n_{\overline{kF}} + \beta_{\overline{k}F})}{B(\alpha_{\overline{k}F}, \beta_{\overline{k}F})} \tag{1.5}$$

$$l_{Fk} = \frac{B(n_F + \alpha_F, n_{\overline{F}} + \beta_F)}{B(\alpha_F, \beta_F)} \tag{1.6}$$

$B(a,b)$ is the *Beta* function which is defined as $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$, and α_{kF}, $\alpha_{k\overline{F}}$, α_F, β_{kF}, $\beta_{k\overline{F}}$, β_F are constants set by the practitioner. In our experiments we set them to be $\alpha_F = 0.2$, $\beta_F = 2/25$ for all words F, $\alpha_{kF} = 0.1$, $\alpha_{k\overline{F}} = 0.1$, $\beta_{kF} = 1/25$ and $\beta_{k\overline{F}} = 1/25$ for all categories k and feature F. These settings correspond to rather diffuse priors.

We explain this score in the context of a two-candidate-word model. The likelihoods for each model and category k are given by

$$M_{(1,1)} : \prod_{i=1}^{M} Pr(x_{i1}, x_{i2}, y_i|\theta_{1k}, \theta_{2k}) = \prod_{i=1}^{M} \mathcal{B}(x_{i1}, \theta_{k1})\mathcal{B}(x_{i1}, \theta_{\overline{k}1})\mathcal{B}(x_{i2}, \theta_{k2}) \times \mathcal{B}(x_{i2}, \theta_{\overline{k}2})Pr(y_i|\theta_k)$$

$$M_{(1,0)} : \prod_{i=1}^{M} Pr(x_{i1}, x_{i2}, y_i|\theta_{1k}, \theta_2) = \prod_{i=1}^{M} \mathcal{B}(x_{i1}, \theta_{k1})\mathcal{B}(x_{i1}, \theta_{\overline{k}1})\mathcal{B}(x_{i2}, \theta_2) \times \mathcal{B}(x_{i2}, \theta_2)Pr(y_i|\theta_k)$$

$$M_{(0,1)} : \prod_{i=1}^{M} Pr(x_{i1}, x_{i2}, y_i|\theta_1, \theta_{2k}) = \prod_{i=1}^{M} \mathcal{B}(x_{i1}, \theta_1)\mathcal{B}(x_{i1}, \theta_1)\mathcal{B}(x_{i2}, \theta_{k2}) \times \mathcal{B}(x_{i2}, \theta_{\overline{k}2})Pr(y_i|\theta_k)$$

$$M_{(0,0)} : \prod_{i=1}^{M} Pr(x_{i1}, x_{i2}, y_i|\theta_1, \theta_2) = \prod_{i=1}^{M} \mathcal{B}(x_{i1}, \theta_1)\mathcal{B}(x_{i1}, \theta_1)\mathcal{B}(x_{i2}, \theta_2) \times \mathcal{B}(x_{i2}, \theta_2)Pr(y_i|\theta_k)$$

where x_{ij} takes the value 1 if document i contains word F_j and 0 otherwise, y_i is 1 if document i is in category k otherwise is 0, $Pr(y_i|\theta_k) = \mathcal{B}(y_i, \theta_k)$ and $\mathcal{B}(x, \theta) = \theta^x(1-\theta)^{1-x}$ denotes a Bernoulli probability distribution.

Therefore, in model $M_{(1,1)}$ the presence or absence of both words in a given document depends on the document class. θ_{k1} corresponds to the proportion of documents in category k with word F_1 and $\theta_{\overline{k}1}$ to the proportion of documents not in category k with word F_1. In model $M_{(1,0)}$ only word F_1 depends on the category of the document and θ_2 correspond to the proportion of documents with word F_2 regardless of the category associated with them. θ_k is the proportion of documents in category k and $Pr(y_i|\theta_k)$ is the probability that document i is in category k.

We assume the following prior probability distributions for the parameters, $\theta_{kF} \sim Beta(\alpha_{kF}, \beta_{kF})$, $\theta_{\overline{k}F} \sim Beta(\alpha_{\overline{k}F}, \beta_{\overline{k}F})$, $\theta_F \sim Beta(\alpha_F, \beta_F)$ and $\theta_k \sim Beta(\alpha_k, \beta_k)$, where $Beta(\alpha, \beta)$ denotes a Beta distribution, i.e. $Pr(\theta|\alpha, \beta) = \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$, $k \in \{1, ..., C\}$ and $F \in \{F_1, ..., F_N\}$.

Then the marginal likelihoods for each of the four models above can be expressed as the products of three terms,

$$Pr(data|M_{(1,1)}) = l_0 \times l_{0F_1k} \times l_{0F_2k}$$
$$Pr(data|M_{(1,0)}) = l_0 \times l_{0F_1k} \times l_{F_2k}$$
$$Pr(data|M_{(0,1)}) = l_0 \times l_{F_1k} \times l_{0F_2k}$$
$$Pr(data|M_{(0,0)}) = l_0 \times l_{F_1k} \times l_{F_2k}$$

where l_{0Fk} and l_{Fk} are defined in equations 1.4 for $F \in \{F_1, F_2, ..., F_N\}$ and l_0 is defined as

$$l_0 = \int_0^1 \prod_i Pr(y_i|\theta_k) Pr(\theta_k|\alpha_k, \beta_k) d\theta_k, \tag{1.7}$$

which is the marginal probability for the category of the documents.

It is straightforward to show that $PIP(F, k)$ in equation 1.2 is equivalent to $PIP(F, k)$ in equation 1.3 if we assume that the prior probability density for the models is uniform, e.g. $Pr(M_l) \propto 1$.

In the example above, the posterior inclusion probability for feature F_1 is given by,

$$\begin{aligned} Pr(F_1|y_k) &= Pr(M_{(1,1)}|data) + Pr(M_{(1,0)}|data) \\ &= \frac{l_{0F_1k}}{l_{0F_1k} + l_{F_1k}}. \end{aligned}$$

To get a single "bag of words" for all categories we compute the weighted average of $PIP(F, k)$ over all categories.

$$PIP(F) = \sum_k Pr(y = k) PIP(F, k).$$

We note that [3] present similar manipulations of the naive Bayes model but for model averaging purposes rather than finding the median probability model.

1.2.3 Posterior Inclusion Probability (PIPp) under Poisson distributions

A generalization of the binary naive Bayes model assumes class-conditional Poisson distributions for the word frequencies in a document. As before, assume that the probability distribution for a word in a document might or might not depend on the category of the document. More precisely, if the distribution for feature x depends on the category k of the document we have

$$\begin{aligned} Pr(x|y = k) &= \frac{e^{-\lambda_{kF}} \lambda_{kF}^x}{x!} \\ Pr(x|y \neq k) &= \frac{e^{-\lambda_{\overline{k}F}} \lambda_{\overline{k}F}^x}{x!} \end{aligned}$$

where x is the number of times word F appears in the document and λ_{kF} ($\lambda_{\overline{k}F}$) represents the expected number of times word F appears in documents in category k ($\overline{k}$). If the distribution for x does not depend on the category of the document we then have

$$Pr(x) = \frac{e^{-\lambda_F} \lambda_F^x}{x!}$$

where λ_F represents the expected number of times word F appears in a document regardless of the category of the document.

Assume the following conjugate prior probability densities for the parameters,

$$\begin{aligned}\lambda_{kF} &\sim Gamma(\alpha_{kF}, \beta_{kF})\\ \lambda_{\overline{k}F} &\sim Gamma(\alpha_{\overline{k}F}, \beta_{\overline{k}F})\\ \lambda_F &\sim Gamma(\alpha_F, \beta_F)\end{aligned}$$

where $\alpha_{kF}, \beta_{kF}, \alpha_{\overline{k}F}, \beta_{\overline{k}F}, \alpha_F, \beta_F$ are hyperparameters to be set by the practitioner.

Now, as before, the posterior inclusion probability for Poisson distributions (PIPp) is given by

$$PIPp(F,k) = \frac{l_{0Fk}}{l_{0Fk} + l_{Fk}}$$

where,

$$\begin{aligned}l_{0Fk} = &\frac{\Gamma(N_{kF}+\alpha_{kF})}{\Gamma(\alpha_{kF})\beta_{kF}^{\alpha_{kF}}}\frac{\Gamma(N_{\overline{k}F}+\alpha_{\overline{k}F})}{\Gamma(\alpha_{\overline{k}F})\beta_{\overline{k}F}^{\alpha_{\overline{k}F}}}\\ &\times(\frac{\beta_{kF}}{n_k\beta_{kF}+1})^{n_{kF}+\alpha_{kF}}(\frac{\beta_{\overline{k}F}}{n_{\overline{k}}\beta_{\overline{k}F}+1})^{n_{\overline{k}F}+\alpha_{\overline{k}F}}\\ l_{Fk} = &\frac{\Gamma(N_F+\alpha_F)}{\Gamma(\alpha_F)}(\frac{\beta_F}{\beta_F n+1})^{n_F+\alpha_F}\frac{1}{\beta_F^{\alpha_F}}.\end{aligned}$$

This time, $N_{kF}, N_{\overline{k}F}, N_F$ denote:
N_{kF}: n° of times word F appears in documents in class k.
$N_{\overline{k}F}$: n° of times word F appears in documents not in class k.
N_F: total n° of times that word F appears in all documents.

As before, to get a single "bag of words" for all categories we compute the weighted average of $PIPp(F,k)$ over all categories,

$$PIPp(F) = \sum_k^C Pr(y=k)PIPp(F,k).$$

1.2.4 Information Gain (IG)

Information gain is a popular score for feature selection in the field of machine learning. In particular it is used in the C4.5 decision tree inductive algorithm. [16] compare five different feature selection scores on two datasets and show that Information Gain is among the two most effective ones. The

information gain of word F is defined to be:

$$\begin{aligned} IG(F) = &- \sum_{k=1}^{C} Pr(y=k) \log Pr(y=k) \\ &+ Pr(F) \sum_{k=1}^{C} Pr(y=k|F) \log Pr(y=k|F) \\ &+ Pr(\overline{F}) \sum_{k=1}^{C} Pr(y=k|\overline{F}) \log Pr(y=k|\overline{F}) \end{aligned}$$

where $\{1, \ldots, C\}$ is the set of categories and $\overline{F}$ the absence of word F. It measures the decrease in entropy when the feature is present versus when the feature is absent.

1.2.5 Bi-Normal Separation (BNS)

The Bi-Normal Separation score, introduced by [5], is defined as:

$$BNS(F,k) = |\Phi^{-1}(\frac{n_{kF}}{n_k}) - \Phi^{-1}(\frac{n_{\overline{k}F}}{n_{\overline{k}}})|$$

where Φ is the standard normal distribution and Φ^{-1} its corresponding inverse. $\Phi^{-1}(0)$ is set to be equal to 0.0005 to avoid numerical problems following [5]. By averaging over all categories, we get a score that selects a single set of words for all categories.

$$BNS(x) = \sum_{k=1}^{C} Pr(y=k)|\Phi^{-1}(\frac{n_{kF}}{n_k}) - \Phi^{-1}(\frac{n_{\overline{k}F}}{n_{\overline{k}}})|$$

To get an idea for what this score is measuring, assume that the probability that a word F is contained in a document is given by $\Phi(\delta_k)$ if the document belongs to class y_k and otherwise is given by $\Phi(\delta_{\overline{k}})$. A word will discriminate with high accuracy between a document that belongs to a category from one that does not, if the value of δ_k is small and the value of $\delta_{\overline{k}}$ is large , or vice versa, if δ_k is large and $\delta_{\overline{k}}$ is small. Now, if we set $\delta_k = \Phi^{-1}(\frac{n_{kF}}{n_k})$ and $\delta_{\overline{k}} = \Phi^{-1}(\frac{n_{\overline{k}F}}{n-n_k})$, the Bi-Normal Separation score is equivalent to the distance between these two quantities, $|\delta_{\overline{k}} - \delta_k|$.

1.2.6 Chi-Square

The chi-square feature selection score, $\chi^2(F,k)$, measures the dependence between word F and category k. If word F and category k are independent $\chi^2(F,k)$ is equal to zero. When we select a different set of words for each category we utilise the following score,

$$\chi^2(F,k) = \frac{n(n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}})^2}{n_k n_F n_{\overline{k}} n_{\overline{F}}}.$$

Again, by averaging over all categories we get a score for selecting a single set of words for all categories.

$$\chi^2(F) = \sum_{k=1}^{C} Pr(y=k)\chi^2(F,k).$$

1.2.7 Odds Ratio

The Odds Ratio measures the odds of word F occuring in documents in category k divided by the odds of word F not occuring in documents in category k. [13] find this to be the best score among eleven scores for a Naive Bayes classifier. For category k and word F the oddsRatio is given by,

$$OddsRatio(F,k) = \frac{\frac{n_{kF}+0.1}{n_k+0.1} / \frac{n_{k\overline{F}}+0.1}{n_k+0.1}}{\frac{n_{\overline{k}F}+0.1}{n_{\overline{k}}+0.1} / \frac{n_{\overline{kF}}+0.1}{n_{\overline{k}}+0.1}}$$

where we add the constant 0.1 to avoid numerical problems. By averaging over all categories we get,

$$OddsRatio(F) = \sum_{k=1}^{C} Pr(y=k)OddsRatio(F,k).$$

1.2.8 Word Frequency

This is the simplest of the feature selection scores. In the study of [16] they show that word frequency is the third best after information gain and χ^2. They also point out that there is strong correlation between these two scores and word frequency. For each category k, word frequency (WF) for word F, is the number of documents in category k that contain word F, i.e.
$WF(F,k) = n_{kF}$.

Averaging over all categories we get a score for each F,

$$WF(F) = \sum_{k=1}^{C} Pr(y=k)WF(F,k) = \sum_{k=1}^{C} Pr(y=k)n_{kF}.$$

1.3 Classification Algorithms

To determine the performance of the different feature selection scores, the classification algorithms that we consider are the Multinomial, Poisson and Binary Naive Bayes classifiers (e.g. [11], [9], and [4]) and the hierarchical probit classifier of [6]. We choose these classifiers for our analysis for two reasons. The first one is the different nature of the classifiers. The naive Bayes models are generative models while the probit is a discriminative model. Generative classifiers learn a model of the joint probability $Pr(x, y)$, where x is the input and y the label. These classifiers make their predictions by using Bayes rule to calculate $Pr(y|x)$. In contrast, discriminative classifiers model the conditional probability of the label given the input ($Pr(y|x)$) directly. The second reason is the good performance that they achieve. In [4] the multinomial model, notwithstanding its simplicity, achieved the best performance among four Naive Bayes models. The hierarchical probit classifier of [6] achieves state of the art performance, comparable to the performance of the best classifiers such as SVM ([8]). We decide to include the binary and Poisson naive Bayes models (see [4] for details) because they allow us to incorporate information of the probability model used to fit the categories of the documents into the feature selection score. For instance, in the Binary Naive Bayes classifiers the features that one can select using the PIP score correspond exactly to the features with the highest posterior inclusion probability. We want to examine whether or not that offers an advantage over other feature selection scores.

1.4 Experimental Settings and Results

Before we start the analysis we remove common noninformative words taken from a standard *stopword* list of 571 words and we remove words that appear less than three times in the training documents. This eliminates $8,752$ words in the Reuters dataset (38% of all words in training documents) and $47,118$ words in the Newsgroups dataset (29% of all words in training documents). Words appear on average in 1.41 documents in the Reuters dataset and in 1.55 documents in the Newsgroups dataset.

We use F1 to measure the performance of the different classifiers and feature selection scores. F1 is the harmonic mean between recall and precision. We average the F1 scores across all categories to get a single value. The micro F1 is a weighted average, where the weights for each category are proportional to the frecuency of documents in the category. The macro F1 gives equal weight to all categories.

1.4.1 Datasets

The 20-Newsgroups dataset contains 19,997 articles divided almost evenly into 20 disjoint categories. The categories topics are related to computers, politics, religion, sport and science. We split the dataset randomly into 75% for training and 25% for testing. We took this version of the dataset from http://www.ai.mit.edu/people/jrennie/20Newsgroups/. Another dataset that we use comes from the Reuters-21578 news story collection. We use a subset of the ModApte version of the Reuters−21,578 collection, where each document has assigned at least one topic label (or category) and this topic label belongs to any of the 10 most populous categories - earn, acq, grain, wheat, crude, trade, interest, corn, ship, money-fx. It contains 6,775 documents in the training set and 2,258 in the testing set.

1.4.2 Experimental Results

In these experiments we compare seven feature selection scores, on two benchmark datasets, Reuters-21578 and Newgroups (see Section 1.4.1), under four classification algorithms (see Section 1.3).

We compare the performance of the classifiers for different numbers of words and vary the number of words from 10 to 1000. For larger number of words the classifiers tend to perform somewhat more similarly, and the effect of choosing the words using a different feature selection procedure is less noticeable.

Figures 1.2- 1.5 show the micro average F1 measure for each of the feature selection scores as we vary the number of features selected for the four classification algorithms we considered: multinomial, probit, Poisson and binary respectively.

We noticed that PIP gives, in general, high values to all very frequent words. To avoid that bias we remove words that appear more than 2000 times in the Reuters dataset (that accounts for 15 words) and more than 3000 times in the Newsgroups dataset (that accounts for 36 words). We now discuss the results for the two datasets:

Reuters. Like the results of [5], if for scalability reasons one is limited to a small number of features (< 50) the best available metrics are IG and χ^2, as Figures 1.2-1.5 show. For larger number of features (> 50), Figure 1.2 shows that PIPp and PIP are the best scores for the mutinomial classifier. Figures 1.4 and 1.5 show the performance for the Poisson and binary classifiers. PIPp and BNS achive the best performance in the Poisson classifier and PIPp achieves the best performance in the binary classifier. WF performs poorly compared to the other scores in all the classifiers, achieving the best performance with the Poisson one.

Newsgroups. χ^2 followed by BNS, IG and PIP are the best performing measures in the probit classifier. χ^2 is also the best one in the multinomial model, followed by BNS and the binary classifier with the macro F1 measure. OR performs best in the Poisson classifier. PIPp is best in the binary classifier

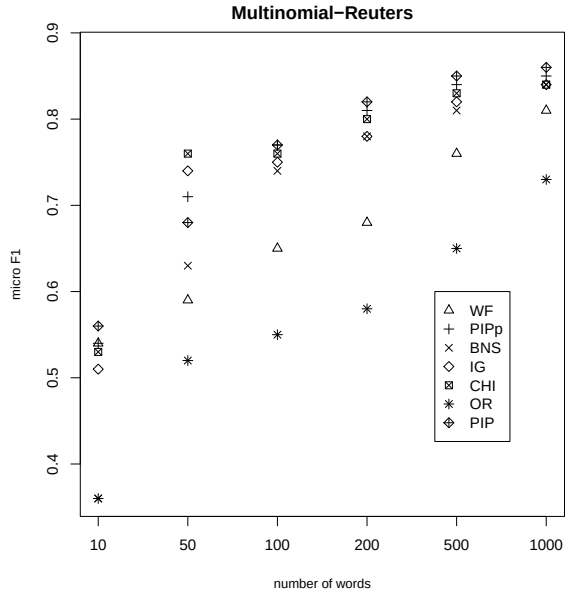

FIGURE 1.2: Performance (for the multinomial model) for different number of words measure micro F1 for the Reuters dataset.

under the micro F1 measure. WF performs poorly compared to the other scores in all classifiers. Because of lack of space we do not show a graphical display of the performance of the classifiers in the Newsgroups dataset, and only the micro F1 measure is displayed graphically for the Reuters dataset.

In Table 1.3 and Table 1.5 we summarize the overall performance of the feature selection scores considered by integrating the curves formed when the dots depicted in Figures 1.2-1.5 are joined. Each column corresponds to a given feature selection. For instance, the number 812 under the header "Multinomial model Reuters-21578" and the row "micro F1" corresponds to the area under the IG "curve" in Figure 1.2. In seven out of sixteen instances χ^2 is the best performing score and in three it is the second best. PIPp in four out of sixteen is the best score and in six is the second best. BNS is the best in two and second best in six. In bold are the best two performance scores.

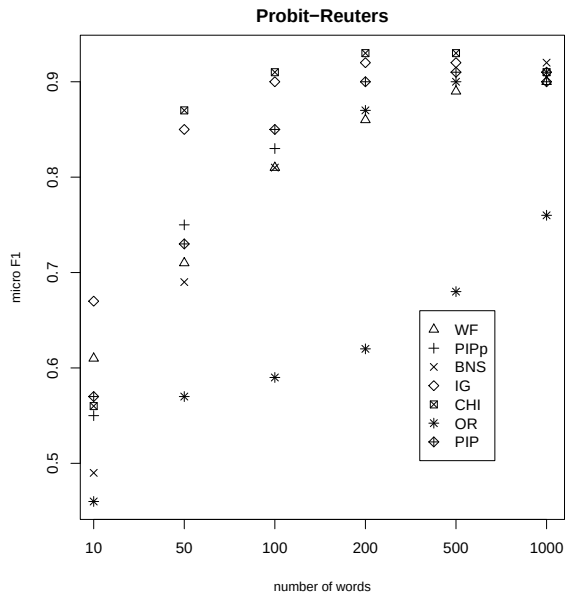

FIGURE 1.3: Performance (for the probit model) for different number of words measure by micro F1 for the Reuters dataset.

1.5 Conclusion

In this study we introduced a new feature selection score, PIP. The value that this score assigns to each word has an appealing Bayesian interpretation, being the posterior probability of inclusion of the word in a naive Bayes model. Such models assume a probability distribution on the words of the documents. We consider two probability distributions, Bernoulli and Poisson. The former takes into account the presence or absence of words in the documents, and the latter, the number of times each word appears in the documents. Future research could consider alternative PIP scores corresponding to different probabilistic models.

χ^2, PIPp, and BNS are the best performing scores. Still, feature selection scores and classification algorithms seem to be highly data- and model-

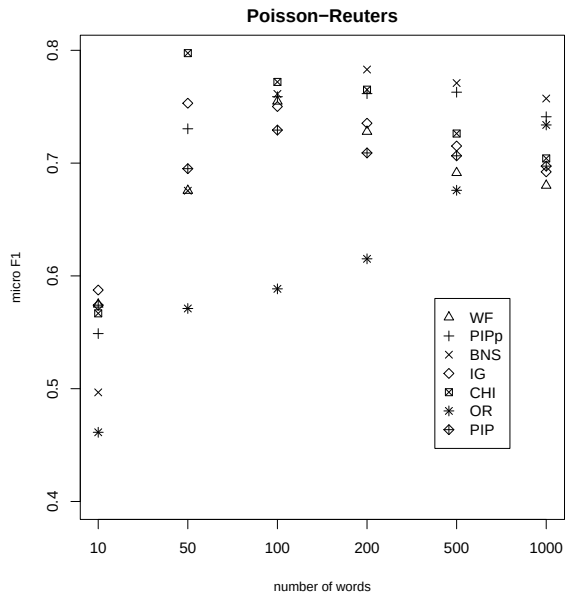

FIGURE 1.4: Performance (for the poisson model) for different number of words measure by micro F1 for the Reuters dataset.

dependent. The feature selection literature reports similarly mixed findings. For instance, [16] find that IG and χ^2 are the strongest feature selection scores. They perform their experiments on two datasets, Reuters-22173 and OHSUMED, and under two classifiers, kNN and a linear least square fit. [13] find that OR is the strongest feature selection score. They perform their experiments on a Naive Bayes model and use the Yahoo dataset. [16] favors bi-normal separation.

Our results regarding the performance of the different scores are consistent with [16] in that χ^2 and IG seem to be strong scores for feature selection in discriminative models, but disagree in that WF appears to be a weak score in most instances. Note that we do not use exactly the same WF score. Ours is a weighted average by the category proportion.

The so-called wrapper approach for feature selection provide an advantage over the filtering approach. The wrapper approach attempts to identify the best feature subset to use with a particular algorithm and dataset, whereas

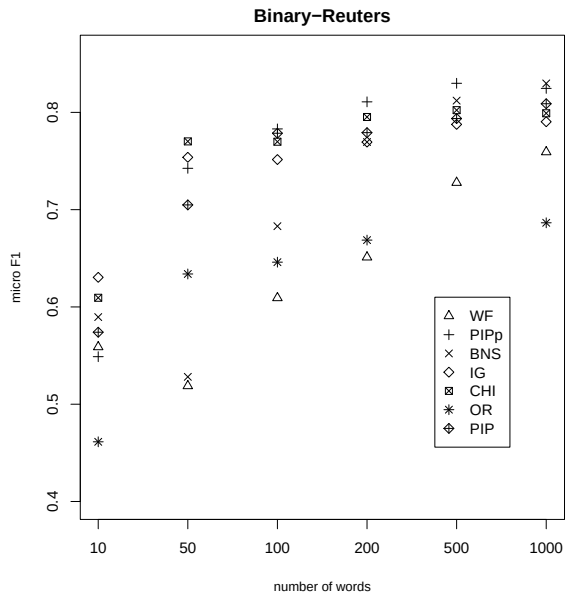

FIGURE 1.5: Performance (for the binary naive Bayes model) for different number of words measure by micro F1 for the Reuters dataset.

the filtering approach attempts to assess the merits of features from the data alone. The feature selection PIP offers that advantage over feature selection scores that follow the filtering approach, for some classifiers. Specifically, for some naive Bayes models like the Binary naive model or Poisson naive model, the score computed by PIP Bernoulli and PIP Poisson depends on the classification algorithm. Our empirical results do not corroborate the benefit of using the same model in the feature selection score and in the classification algorithm. The strong assumption that naive Bayes models make about the independence of the features given the label, is well known not to be suitable for textual datasets, words tend to be correlated. Despite the correlation structure of words, naive Bayes classifiers have been shown to give highly accurate predictions. The reason for that are clearly explained in [7]. The authors are currently exploring extensions of this method of feature selection to applications where the naive Bayes assumption appears to be more suitable.

TABLE 1.2: Performance of the Binary and Poisson models

	IG	χ^2	OR	BNS	WF	PIP	PIPp
	Poisson model Reuters-21578 dataset						
micro F_1	708	719	670	**763**	684	699	**755**
macro F_1	618	**628**	586	**667**	590	618	**667**
	Poisson model 20-Newsgroups dataset						
micro F_1	753	808	**928**	812	684	777	**854**
macro F_1	799	841	**936**	841	773	813	**880**
	Berboulli model Reuters-21578 dataset						
micro F_1	779	794	669	**804**	721	786	**822**
macro F_1	680	**698**	618	709	614	696	**746**
	Bernoulli model 20-Newsgroups dataset						
micro F_1	531	**566**	508	556	436	534	**650**
macro F_1	628	**673**	498	**652**	505	627	650

FIGURE 1.5: This table summarizes an overall performance of the feature selection scores considered by integrating the curves formed by joining the dots depicted in Figures 1.2-1.5. In bold are the best two performing score.

TABLE 1.4: Performance of the Multinomial and Probit models

	IG	χ^2	OR	BNS	WF	PIP	PIPp
	Multinomial model Reuters-21578 dataset						
micro F_1	812	822	644	802	753	**842**	**832**
macro F_1	723	733	555	713	644	**762**	**753**
	Multinomial model 20-Newsgroups dataset						
micro F_1	535	**614**	575	**584**	456	564	575
macro F_1	594	**644**	565	**634**	486	604	585
	Probit model Reuters-21578 dataset						
micro F_1	**911**	**921**	674	891	881	901	891
macro F_1	**861**	**861**	605	842	753	842	**851**
	Probit model 20-Newsgroups dataset						
micro F_1	703	**723**	575	**713**	565	693	644
macro F_1	693	**723**	565	**703**	565	683	624

FIGURE 1.5: This table summarizes an overall performance of the feature selection scores considered by integrating the curves formed by joining the dots depicted in Figures 1.2-1.5. In bold are the best two performing score.

References

[1] Barbieri, M.M. and Berger, J.O. Optimal predictive model selection. *Annals of Statistics*, **32**, 870–897, 2004.

[2] Bernardo, J. M. and Smith, A. F. M. *Bayesian Theory*. New York: Wiley, 1994.

[3] Dash, D. and Cooper, G.F. Exact model averaging with naive Bayesian classifiers. In: *Proceedings of the Nineteenth International Conference on Machine Learning*, 91-98, 2002.

[4] Eyheramendy, S., Lewis, D.D. and Madigan, D. On the naive Bayes classifiers for text categorization. In *Proceedings of the ninth international workshop on Artificial Intelligence and Statistics*, eds, C.M. Bishop and B.J. Frey, 2003.

[5] Forman, G. An extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 2003.

[6] Genkin, A., Lewis, D.D., Eyheramendy, S., Ju, W.H. and Madigan, D. Sparse Bayesian Classifiers for Text Categorization, submitted to JICRD, 2003.

[7] Hand, D.J. and Yu, K. Idiot's Bayes Not so Stupid after All? International Statistical Review, vol. 69, no. 3, pp. 385-398, 2001.

[8] Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of ECML-98, 137-142, 1998.

[9] Lewis, D.D. Naive (Bayes) at forty: The independence assumption in information retrieval. Proceedings of ECML-98, 4–15, 1998.

[10] Liu, H. and Yu, L. Towards integrating feature selection algorithms for classification and clustering. IEEE Transactions in knowledge and data ingeneering, 491-502, 2005.

[11] McCallum, A. and Nigam, K. A comparison of event models for naive Bayes text classification. In *AAAI/ICML Workshop on Learning for Text Categorization*, pages 41 – 48, 1998.

[12] Miller, A.J. *Subset selection in regression (second edition)*. Chapman and Hall, 2002.

0-8493-0052-5/00/$0.00+$.50

[13] Mladenic, D. and Grobelnik, M. Feature selection for unbalanced class distribution and naive Bayes. Proceedings ICML-99, pages 258-267, 1999.

[14] Pardi, F., Lewis, C.M. and Whittaker, J.C. SNP selection for association studies: maximizing power across SNP choice and study size Ann Human Genetics

[15] Silvey, S. D. Statistical Inference. Chapman & Hall. London, 1975.

[16] Yang, Y. and Pedersen, J.O. A comparative study on feature selection in text categorization. Proceedings ICML-97, 412-420, 1997.